**ROAD ACCIDENT ANALYSIS USING PYTHON IN POWER BI**

**Count of Fatal Casualities:**

import matplotlib.pyplot as plt

import matplotlib as mpl

# Adjust font size

mpl.rcParams['font.size'] = 50

# Sample data

numbers = dataset.loc[0:307974, 'CY\_Casualties'].tolist()

names = dataset.loc[0:307974, 'Fatal (groups)'].tolist()

# Define colors

colors = [(255/255, 54/255, 121/255), (0/255, 185/255, 170/255)]

# Create pie chart

plt.figure(figsize=(8, 8))  # Adjust figure size if needed

plt.pie(numbers, labels=names, autopct='%1.1f%%', startangle=140, pctdistance=0.85, colors=colors)

# Draw circle

centre\_circle = plt.Circle((0, 0), 0.70, fc='white')

fig = plt.gcf()

fig.gca().add\_artist(centre\_circle)

# Equal aspect ratio ensures that pie is drawn as a circle

plt.axis('equal')

# Title

plt.title('Fatal')

# Show plot

plt.show()

**Count of Serious Casualities:**

import matplotlib.pyplot as plt

import matplotlib as mpl

# Adjust font size

mpl.rcParams['font.size'] = 50

# Sample data

numbers = dataset.loc[0:307974, 'CY\_Casualties'].tolist()

names = dataset.loc[0:307974, 'Serious(groups)'].tolist()

# Define colors

colors = [(0/255, 185/255, 170/255),(255/255, 54/255, 121/255)]

# Create pie chart

plt.figure(figsize=(8, 8))  # Adjust figure size if needed

plt.pie(numbers, labels=names, autopct='%1.1f%%', startangle=140, pctdistance=0.85, colors=colors)

# Draw circle

centre\_circle = plt.Circle((0, 0), 0.70, fc='white')

fig = plt.gcf()

fig.gca().add\_artist(centre\_circle)

# Equal aspect ratio ensures that pie is drawn as a circle

plt.axis('equal')

# Title

plt.title('Serious')

# Show plot

plt.show()

**Count of Slight Casualities:**

import matplotlib.pyplot as plt

import matplotlib as mpl

# Adjust font size

mpl.rcParams['font.size'] = 50

# Sample data

numbers = dataset.loc[0:307974, 'CY\_Casualties'].tolist()

names = dataset.loc[0:307974, 'Slight(groups)'].tolist()

# Define colors

colors = [(0/255, 185/255, 170/255),(255/255, 54/255, 121/255)]

# Create pie chart

plt.figure(figsize=(8,8))  # Adjust figure size if needed

plt.pie(numbers, labels=names, autopct='%1.1f%%', startangle=140, pctdistance=0.85, colors=colors)

# Draw circle

centre\_circle = plt.Circle((0, 0), 0.70, fc='white')

fig = plt.gcf()

fig.gca().add\_artist(centre\_circle)

# Equal aspect ratio ensures that pie is drawn as a circle

plt.axis('equal')

# Title

plt.title('Slight')

# Show plot

plt.show()

**Light\_Condition and Weather\_Conditions by Casualities:**

import matplotlib.pyplot as plt

import numpy as np

import matplotlib as mpl

# Define your data (replace 'dataset' with your actual dataset)

mpl.rcParams['font.size'] = 20.0

weather\_data = dataset.groupby(['Weather\_Conditions (groups)', 'Light\_Conditions (groups)'])['CY\_Casualties'].sum().unstack().fillna(0)

# Define the width of each bar

bar\_width = 0.35

# Define the positions for the bars

positions = np.arange(len(weather\_data))

# Create the clustered bar chart

plt.figure(figsize=(12, 6))

for i, column in enumerate(weather\_data.columns):

    # Set colors for bars

    colors = ['#FF3679', '#00B9AA']  # Using hexadecimal color codes

    bars = plt.bar(positions + bar\_width \* i, weather\_data[column], width=bar\_width, label=column, color=colors[i])

    for bar in bars:

        height = bar.get\_height()

        plt.text(bar.get\_x() + bar.get\_width() / 2, height, f'{int(height)}', ha='center', va='bottom')

# Customize labels and title

plt.ylabel('Number of Casualties')

plt.xticks(positions + bar\_width \* (len(weather\_data.columns) - 1) / 2, weather\_data.index, rotation=45)

# Add legend

plt.legend()

# Show the plot

plt.tight\_layout()

plt.show()

**Casualities By Road\_Type:**

# The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:

# dataset = pandas.DataFrame(undefined)

# dataset = dataset.drop\_duplicates()

# Paste or type your script code here:

# The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:

# dataset = pandas.DataFrame(Road\_Type, CY\_Casualties)

# dataset = dataset.drop\_duplicates()

# Paste or type your script code here:

import matplotlib.pyplot as plt

import matplotlib as mb

# Adjust font size

mb.rcParams['font.size'] = 22

# Assuming you have already loaded your dataset into a DataFrame called 'dataset'

# Assuming 'Total\_Casualties' and 'Day\_of\_Week' are columns in your dataset

# Extracting data for plotting

numbers = dataset.loc[0:307974, 'CY\_Casualties'].tolist()

names = dataset.loc[0:307974, 'Road\_Type'].tolist()

# Plotting the horizontal bar chart

plt.figure(figsize=(15, 6))

bars = plt.barh(names, numbers, color='#FF3679', height=0.6)  # Use barh() for horizontal bar chart

plt.grid(axis='x')  # Show gridlines on the x-axis

# Adding tooltips

for bar, name, number in zip(bars, names, numbers):

    plt.text(number, name, str(number), ha='left', va='center')

plt.tight\_layout()  # Adjust layout

plt.show()

**Casualities By Days of Week:**

import matplotlib.pyplot as plt

import matplotlib as mb

# Set the font size

mb.rcParams['font.size'] = 18.0

# Define the order of days of the week

week\_order = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday']

# Map day names to their corresponding numbers

day\_to\_num = {'Sunday': 0, 'Monday': 1, 'Tuesday': 2, 'Wednesday': 3, 'Thursday': 4, 'Friday': 5, 'Saturday': 6}

# Apply mapping and sort the data

sorted\_data = dataset.loc[0:307974].copy()

sorted\_data['Day\_Num'] = sorted\_data['Day\_of\_Week'].map(day\_to\_num)

sorted\_data = sorted\_data.sort\_values(by='Day\_Num')

# Plot the bar chart

plt.figure(figsize=(12, 6))

plt.bar(sorted\_data['Day\_of\_Week'], sorted\_data['CY\_Casualties'], color='#FF3679', width=0.3)  # Set color to RGB(0, 167, 149)

plt.xlabel('Day of Week')

plt.ylabel('CY\_Casualties')

plt.show()

**Casualities By Location:**

import matplotlib.pyplot as plt

import matplotlib as mpl

# Adjust font size

mpl.rcParams['font.size'] = 30.0

# Sample data

numbers = dataset.loc[0:307974, 'CY\_Casualties'].tolist()

names = dataset.loc[0:307974, 'Urban\_or\_Rural\_Area'].tolist()

# Define colors

colors = ['#00B9AA','#FF3679']  # Using hexadecimal color codes

# Create pie chart

plt.figure(figsize=(8, 8))  # Adjust figure size if needed

plt.pie(numbers, labels=names, autopct='%1.1f%%', startangle=140, pctdistance=0.85, colors=colors)

# Equal aspect ratio ensures that pie is drawn as a circle

plt.axis('equal')

# Show plot

plt.show()

**Number\_of\_Casualities by Speed\_Limit:**

# The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:

# dataset = pandas.DataFrame(headcount)

# dataset = dataset.drop\_duplicates()

# Paste or type your script code here:

import matplotlib.pyplot as plt

import seaborn as sns

import matplotlib as mpl

mpl.rcParams['font.size']=20

plt.figure(figsize=(8, 4))

sns.boxplot(data=dataset, y='Number\_of\_Casualties',x='Speed\_limit', color='#00B9AA')

plt.title('Distribution of Number of Casualties')

plt.ylabel('Number of Casualties')

plt.show()